

Optimal Code Design for Lossless and Near Lossless Source Coding in Multiple Access Networks *

Qian Zhao

Michelle Effros

Abstract

A multiple access source code (MASC) is a source code designed for the following network configuration: a pair of correlated information sequences $\{X_i\}_{i=1}^{\infty}$ and $\{Y_i\}_{i=1}^{\infty}$ is drawn i.i.d. according to joint probability mass function (p.m.f.) $p(x, y)$; the encoder for each source operates without knowledge of the other source; the decoder jointly decodes the encoded bit streams from both sources. The work of Slepian and Wolf describes all rates achievable by MASCs with arbitrarily small but non-zero error probabilities but does not address truly lossless coding or code design. In this paper, we consider practical code design for lossless and near lossless MASCs. We generalize the Huffman and arithmetic code design algorithms to attain the corresponding optimal MASC codes for arbitrary p.m.f. $p(x, y)$. Experimental results comparing the optimal achievable rate region to the Slepian-Wolf region are included.

I Introduction

A multiple access network is a system with several transmitters sending information to a single receiver. One example of a multiple access system is a sensor network, where a collection of separately located sensors sends correlated information to a central processing unit. MASCs yield efficient data representations for multiple access systems when cooperation among the transmitters is not possible.

In multiple access source coding, correlated information sequences $\{X_i\}_{i=1}^{\infty}$ and $\{Y_i\}_{i=1}^{\infty}$ are drawn i.i.d. according to joint p.m.f. $p(x, y)$. The encoder for each source operates without knowledge of the other source. The decoder receives the encoded bit streams from both sources. The results of [1] describe the rates achievable in this scenario with arbitrarily small (but non-zero) probability of decoding error and arbitrarily large coding dimension. Making these ideas applicable in practice requires MASC design algorithms for finite dimensions. This work treats the design of both truly lossless and near lossless MASCs. (Near lossless codes are desirable for use in lossy MASCs.) Prior works on practical lossless MASCs include [2, 3, 4, 5]. Earlier code designs are optimal for at most a restricted class of source distributions.

In this work, we generalize the optimal Huffman and arithmetic code design algorithms from the traditional single-sender, single-receiver source coding scenario to the MASC scenario. We address the problem in two stages. First, we consider the special

*Q. Zhao(Tel: 626-395-2218, qianz@caltech.edu) and M. Effros(effros@caltech.edu) are with the Department of Electrical Engineering, MC 136-93, California Institute of Technology, Pasadena, CA 91125. This material is based upon work supported by NSF under Award No. CCR-9909026.

case where X is described using a traditional source code and the goal is to minimize the expected description length for Y given the decoder's knowledge of X . This is equivalent to the problem of losslessly describing Y in the presence of side information available only to the decoder; hence we refer to this problem as "side-information coding." In Section II, we generalize the Huffman and arithmetic coding algorithms to design optimal lossless side-information codes. In Section III, we expand on the ideas of Section II to achieve optimal lossless and near lossless codes for the general MASC configuration. Sections IV and V contain experimental results and a summary of this paper's key contributions.

II Lossless Side-Information Coding

We consider finite-alphabet memoryless sources $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ with joint p.m.f. $p(x, y)$. A lossless instantaneous MASC for (X, Y) consists of two encoders $\gamma_X : \mathcal{X} \rightarrow \{0, 1\}^*$ and $\gamma_Y : \mathcal{Y} \rightarrow \{0, 1\}^*$ and a decoder $\gamma^{-1} : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \mathcal{X} \times \mathcal{Y}$. Here $\gamma_X(x)$ and $\gamma_Y(y)$ are the binary descriptions of x and y , $\gamma^{-1}(\gamma_X(x), \gamma_Y(y)) = (x, y)$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and for any x_1, x_2, \dots and y_1, y_2, \dots the instantaneous decoder reconstructs (x_1, y_1) by reading only the first $|\gamma_X(x_1)|$ bits from $\gamma_X(x_1)\gamma_Y(y_1)\dots$ and the first $|\gamma_Y(y_1)|$ bits from $\gamma_Y(y_1)\gamma_X(x_1)\dots$ (without prior knowledge of these lengths).

In the side-information case, we treat the design of γ_Y when the decoder knows X . This describes an MASC where γ_X encodes X using a traditional code for p.m.f. $\{p(x)\}$ and γ_Y encodes Y assuming that the decoder decodes X before decoding Y .

One class of lossless instantaneous side-information codes is introduced in [3] and [5]. Here source Y is encoded so that $y, y' \in \mathcal{A}_x = \{y \in \mathcal{Y} : p(x, y) > 0\}$ for some $x \in \mathcal{X}$ implies that $\gamma_Y(y)$ is not a prefix of $\gamma_Y(y')$. The decoder first losslessly decodes X and then uses the value of X to determine the set $\{\gamma_Y(y) : y \in \mathcal{A}_X\}$ from which to decode Y . Since these codewords satisfy the prefix condition, the description of $Y \in \mathcal{A}_X$ is uniquely decodable given X .¹ While [3] and [5] lend insight into lossless side-information coding, neither provides an optimal design algorithm.

In this paper, we describe a constructive algorithm for designing optimal lossless side-information codes for an arbitrary finite alphabet $\mathcal{X} \times \mathcal{Y}$ and p.m.f. $p(x, y)$. Since $\mathcal{X} \times \mathcal{Y}$ is arbitrary, \mathcal{X} and \mathcal{Y} can be extension alphabets $\mathcal{X} = \tilde{\mathcal{X}}^n$ and $\mathcal{Y} = \tilde{\mathcal{Y}}^n$. Thus our optimal MASC design algorithm for $\mathcal{X} \times \mathcal{Y}$ yields an optimal MASC design algorithm for $\tilde{\mathcal{X}}^n \times \tilde{\mathcal{Y}}^n$ for any n .

Groups and Matched Codes: Definitions and Properties

We begin by developing terminology for describing which symbols from \mathcal{Y} have binary descriptions that are identical and which have binary descriptions that are prefixes of each other. In particular, if we wish to encode two distinct symbols with the same binary description, then we join those symbols together in a "1-level group." If we wish to give one 1-level group a binary description that is a prefix of the binary description of other 1-level groups, then we build a "2-level group"; the 2-level group can be represented by a tree-structure with the first 1-level group sitting at its root and the remaining groups descending from there. These ideas generalize to M -level

¹Both [3] and [5] allow $\gamma_Y(y) = \gamma_Y(y')$ when $y \in \mathcal{A}_x$ implies $y' \notin \mathcal{A}_x$ for all $x \in \mathcal{X}$. However, [3] requires that distinct codewords be prefix-free, ruling out the optimal solution for many $p(x, y)$.

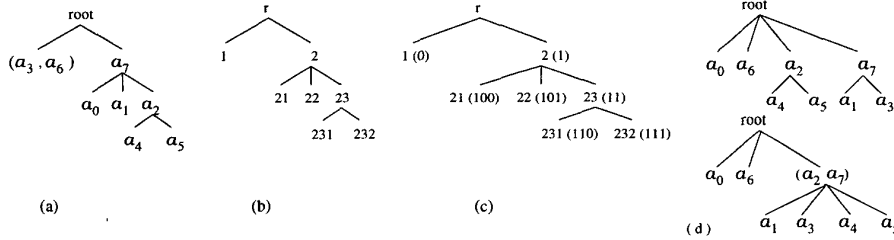


Figure 1: (a) Partition tree $\mathcal{T}(\mathcal{P})$; (b) labels for $\mathcal{T}(\mathcal{P})$; (c) matched code for \mathcal{P} ; (d) Combining groups in partition $\{(a_0), ((a_2) : \{(a_4), (a_5)\}), ((a_7) : \{(a_1), (a_3)\}), (a_6)\}$.

groups with $M > 2$. We define these terms carefully below, ruling out constructions that cannot yield lossless side-information codes. These definitions allow us to design codes for the nested descriptions of groups rather than the description of symbols.

Symbols $y_1, y_2 \in \mathcal{Y}$ can be *combined* under p.m.f. $p(x, y)$ if $p(x, y_1)p(x, y_2) = 0$ for each $x \in \mathcal{X}$. The collection $\mathcal{G} = (y_1, \dots, y_m)$ is called a *1-level group* for $p(x, y)$ if each pair of distinct members $y_i, y_j \in \mathcal{G}$ can be combined under $p(x, y)$. For any $y \in \mathcal{Y}$ and any $p(x, y)$, (y) is a 1-level group. The *tree representation* $\mathcal{T}(\mathcal{G})$ for 1-level group \mathcal{G} is a single node representing all members of \mathcal{G} .

A *2-level group* for p.m.f. $p(x, y)$, denoted by $\mathcal{G} = (\mathcal{R} : \mathcal{C}(\mathcal{R}))$ comprises a root \mathcal{R} and its children $\mathcal{C}(\mathcal{R})$, where \mathcal{R} is a 1-level group, $\mathcal{C}(\mathcal{R})$ is a set of 1-level groups, and for each $\mathcal{G}' \in \mathcal{C}(\mathcal{R})$, each pair $y_1 \in \mathcal{R}$ and $y_2 \in \mathcal{G}'$ can be combined under $p(x, y)$. Members of all $\mathcal{G}' \in \mathcal{C}(\mathcal{R})$ are called members of $\mathcal{C}(\mathcal{R})$, and members of \mathcal{R} and $\mathcal{C}(\mathcal{R})$ are called members of \mathcal{G} . In the tree representation $\mathcal{T}(\mathcal{G})$ for \mathcal{G} , the root node $\mathcal{T}(\mathcal{R})$ is the parent of all subtrees $\mathcal{T}(\mathcal{G}')$, $\mathcal{G}' \in \mathcal{C}(\mathcal{R})$.

For each subsequent $M > 2$, an *M-level group* for $p(x, y)$ is a pair $\mathcal{G} = (\mathcal{R} : \mathcal{C}(\mathcal{R}))$ such that for each $\mathcal{G}' \in \mathcal{C}(\mathcal{R})$, each pair $y_1 \in \mathcal{R}$ and $y_2 \in \mathcal{G}'$ can be combined under p.m.f. $p(x, y)$. Here \mathcal{R} is a 1-level group and $\mathcal{C}(\mathcal{R})$ is a set of groups of $M - 1$ or fewer levels, at least one of which is an $(M - 1)$ -level group. The membership and tree representation of \mathcal{G} are defined in the same way as in a 2-level group.

We use the following $p(x, y)$ with $\mathcal{X} = \mathcal{Y} = \{a_0, a_1, \dots, a_7\}$ as an example. Here

| $x \backslash y$ | a_0 | a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| a_0 | 0.04 | 0 | 0.15 | 0 | 0 | 0 | 0 | 0 |
| a_1 | 0 | 0.04 | 0 | 0.05 | 0.06 | 0 | 0 | 0 |
| a_2 | 0.04 | 0 | 0.05 | 0 | 0 | 0 | 0.01 | 0 |
| a_3 | 0.02 | 0 | 0 | 0.06 | 0 | 0.01 | 0 | 0 |
| a_4 | 0 | 0.05 | 0 | 0 | 0.05 | 0.02 | 0 | 0 |
| a_5 | 0 | 0.1 | 0 | 0 | 0 | 0.03 | 0.06 | 0 |
| a_6 | 0 | 0 | 0 | 0 | 0 | 0 | 0.02 | 0.05 |
| a_7 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0.08 |

$\mathcal{G}(3) = ((a_7) : \{(a_0), (a_1), ((a_2) : \{(a_4), (a_5)\})\})$ is a 3-level group with members $\{a_0, a_1, a_2, a_4, a_5, a_7\}$. The root (a_7) of $\mathcal{T}(\mathcal{G}(3))$ has three children: nodes $\mathcal{T}(a_0)$ and $\mathcal{T}(a_1)$ and a 2-level tree with root node $\mathcal{T}(a_2)$ and children $\mathcal{T}(a_4)$, $\mathcal{T}(a_5)$.

A *partition* \mathcal{P} of \mathcal{Y} for $p(x, y)$ is a complete and non-overlapping set of groups. That is, $\mathcal{P} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$ satisfies $\bigcup_{i=1}^m \mathcal{G}_i = \mathcal{Y}$ and $\mathcal{G}_j \cap \mathcal{G}_k = \emptyset$ for any $j \neq k$,

where each $\mathcal{G}_i \in \mathcal{P}$ is a group for $p(x, y)$, and $\mathcal{G}_j \cup \mathcal{G}_k$ and $\mathcal{G}_j \cap \mathcal{G}_k$ refer to the union and intersection respectively of the members of \mathcal{G}_j and \mathcal{G}_k . The tree representation $\mathcal{T}(\mathcal{P})$ for partition \mathcal{P} , called a *partition tree*, is built by first constructing $\mathcal{T}(\mathcal{G}_i)$ for each i and then linking the root of all $\mathcal{T}(\mathcal{G}_i)$, $i \in \{1, \dots, m\}$ to a single node, which is defined as the root \mathbf{r} of $\mathcal{T}(\mathcal{P})$. Figure 1(a) shows the partition tree for $\mathcal{P} = \{(a_3, a_6), \mathcal{G}(3)\}$.

For any 1-level group \mathcal{G} at depth d in $\mathcal{T}(\mathcal{P})$, let \mathbf{n} describe the d -step path from root \mathbf{r} to node $\mathcal{T}(\mathcal{G})$ in $\mathcal{T}(\mathcal{P})$. We refer to \mathcal{G} by describing this path. To make the path descriptions unique, we fix an order on the descendants of each node and number them from left to right. Thus \mathbf{n} 's children are labeled as $\mathbf{n}1, \mathbf{n}2, \dots, \mathbf{n}K(\mathbf{n})$, where $\mathbf{n}k$ is a vector created by concatenating k to \mathbf{n} and $K(\mathbf{n})$ is the number of children descending from \mathbf{n} . The labeled partition tree for Figure 1(a) appears in Figure 1(b).

A *matched code* γ_Y for partition \mathcal{P} is a binary code such that for any node $\mathcal{T}(\mathbf{n}) \in \mathcal{T}(\mathcal{P})$ and symbols $y_1, y_2 \in \mathbf{n}$ and $y_3 \in \mathbf{n}k$, $k \in \{1, \dots, K(\mathbf{n})\}$: (1) $\gamma_Y(y_1) = \gamma_Y(y_2)$; (2) $\gamma_Y(y_1)$ is a prefix of $\gamma_Y(y_3)$; (3) $\{\gamma_Y(\mathbf{n}k) : k \in \{1, \dots, K(\mathbf{n})\}\}$ is prefix-free. If symbol $y \in \mathcal{Y}$ belongs to 1-level group \mathcal{G} , then $\gamma_Y(y)$ describes the path from \mathbf{r} to $\mathcal{T}(\mathcal{G})$ in $\mathcal{T}(\mathcal{P})$; the path description is a concatenated list of step descriptions, where the step from $\mathcal{T}(\mathbf{n})$ to $\mathcal{T}(\mathbf{n}k)$, $k \in \{1, \dots, K(\mathbf{n})\}$ is described using a prefix-code on $\{1, \dots, K(\mathbf{n})\}$. An example of a matched code for partition $\{(a_3, a_6), \mathcal{G}(3)\}$ appears in Figure 1(c), where the codeword for each node is indicated in parentheses.

In the above framework, a partition specifies the prefix and equivalence relationships in the binary descriptions of $y \in \mathcal{Y}$; a matched code is any code with those properties. Our definitions enforce the condition that for any matched code, $y_1, y_2 \in \mathcal{A}_x$ for some $x \in \mathcal{X}$ implies that $\gamma_Y(y_1)$ is not a prefix of $\gamma_Y(y_2)$; that is, γ_Y violates the prefix property only when knowing X eliminates all possible ambiguity. Theorem 1 establishes the equivalence of matched codes and lossless side-information codes.

Theorem 1 [6, 7] *Code γ_Y is a lossless instantaneous side-information code for $p(x, y)$ if and only if γ_Y is a matched code for some partition \mathcal{P} of \mathcal{Y} for $p(x, y)$.*

Using Theorem 1, we break the problem of lossless side-information code design into two parts: partition design and matched code design. We begin with the second part.

Matched Code Design: Optimal Shannon, Huffman, and Arithmetic Codes

Given an arbitrary partition \mathcal{P} of \mathcal{Y} for $p(x, y)$, we wish to design the optimal matched code for \mathcal{P} . In traditional lossless coding, the optimal description lengths are $l^*(x) = -\log p(x)$ for all $x \in \mathcal{X}$ if those lengths are all integers. Theorem 2 gives the corresponding result for lossless side-information codes on a fixed partition \mathcal{P} .

Theorem 2 [6, 7] *Given partition \mathcal{P} of \mathcal{Y} for $p(x, y)$, the optimal matched code for \mathcal{P} has description lengths $l^*(\mathbf{r}) = 0$ and $l^*(\mathbf{n}k) = l^*(\mathbf{n}) + \log_2(\sum_{j=1}^{K(\mathbf{n})} Q(\mathbf{n}j)/Q(\mathbf{n}k))$ for all $\mathcal{T}(\mathbf{n}) \in \mathcal{T}(\mathcal{P})$ and $k \in \{1, \dots, K(\mathbf{n})\}$ if those lengths are all integers.*

We present three strategies for building matched codes that approximate the optimal length function of Theorem 2. For any node $\mathcal{T}(\mathbf{n})$ with $K(\mathbf{n}) > 0$, the first matched code $\gamma_Y^{(S)}$ describes the step from $\mathcal{T}(\mathbf{n})$ to $\mathcal{T}(\mathbf{n}k)$ using a Shannon code with alphabet $\{1, \dots, K(\mathbf{n})\}$ and p.m.f. $\{Q(\mathbf{n}k)/\sum_{j=1}^{K(\mathbf{n})} Q(\mathbf{n}j)\}_{k=1}^{K(\mathbf{n})}$; the resulting description lengths are $l^{(S)}(\mathbf{r}) = 0$ and $l^{(S)}(\mathbf{n}k) = l^{(S)}(\mathbf{n}) + \lceil \log_2(\sum_{j=1}^{K(\mathbf{n})} Q(\mathbf{n}j)/Q(\mathbf{n}k)) \rceil$. The

second and third matched codes, $\gamma_Y^{(H)}$ and $\gamma_Y^{(A)}$, replace the Shannon codes of $\gamma_Y^{(S)}$ with Huffman and arithmetic codes, respectively, matched to the same p.m.f.s. The “matched Huffman code” $\gamma_Y^{(H)}$ is optimal by Theorem 3.

Theorem 3 [6, 7] *Given a partition \mathcal{P} , the matched Huffman code for \mathcal{P} achieves the optimal expected rate over all matched codes for \mathcal{P} .*

The description length $l^{(A)}(x^n)$ in coding data sequence x^n using a 1-dimensional “matched arithmetic code” $\gamma_Y^{(A)}$ satisfies $l^{(A)}(x^n) \leq \sum_{i=1}^n l^*(x_i) + 2$, giving a normalized description length arbitrarily close to the optimum for n sufficiently large.

Optimal Partitions: Definitions and Properties

The discussion above describes optimal Shannon, Huffman, and arithmetic lossless side-information codes for a given partition \mathcal{P} . The partition \mathcal{P} yielding the best performance remains to be found. We focus on Huffman and arithmetic coding.

Given a partition \mathcal{P} , let $l_{\mathcal{P}}^{(H)}$ and $l_{\mathcal{P}}^*$ be the Huffman and optimal description lengths respectively for \mathcal{P} . We say that \mathcal{P} is *optimal for matched Huffman coding* on $p(x, y)$ if $El_{\mathcal{P}}^{(H)}(Y) \leq El_{\mathcal{P}'}^{(H)}(Y)$ for any other partition \mathcal{P}' of \mathcal{Y} for $p(x, y)$ (and therefore, by Theorems 1 and 3, $El_{\mathcal{P}}^{(H)}(Y) \leq El(Y)$ where l is the description length for any other instantaneous lossless side-information code on $p(x, y)$). We say that \mathcal{P} is *optimal for matched arithmetic coding* on $p(x, y)$ if $El_{\mathcal{P}}^*(Y) \leq El_{\mathcal{P}'}^*(Y)$ for any other partition \mathcal{P}' of \mathcal{Y} on $p(x, y)$. Some properties of optimal partitions follow. Each property holds under either of the above definitions of optimality.

Lemma 1 *There exists an optimal partition \mathcal{P}^* of \mathcal{Y} for $p(x, y)$ for which every node except for the root of \mathcal{P}^* is non-empty.*

Proof. If any non-root node \mathbf{n} of partition \mathcal{P} is empty, then removing \mathbf{n} , so $\{\mathbf{nk}\}_{k=1}^{K(\mathbf{n})}$ descend directly from \mathbf{n} 's parent, gives new partition \mathcal{P}' . Any matched code on \mathcal{P} , including the optimal matched code on \mathcal{P} , is a matched code on \mathcal{P}' . \square

Lemma 2 *If $\mathcal{T}_1, \dots, \mathcal{T}_m$ are the subtrees descending from any node \mathbf{n} in optimal partition \mathcal{P}^* of \mathcal{Y} for $p(x, y)$, then the tree where $\{\mathcal{T}_1, \dots, \mathcal{T}_m\}$ descend from an empty root is identical to $\mathcal{T}(\hat{\mathcal{P}}^*)$, where $\hat{\mathcal{P}}^*$ is an optimal partition of $\hat{\mathcal{Y}} = \cup_{i=1}^m \mathcal{T}_i$ for $p(x, y)$.*

Proof. Since the matched code's description can be broken into a description of \mathbf{n} followed by a matched code on $\{\mathcal{T}_1, \dots, \mathcal{T}_m\}$, the partition described by $\mathcal{T}(\mathcal{P})$ cannot be optimal unless the partition described by $\{\mathcal{T}_1, \dots, \mathcal{T}_m\}$ is. \square

The List of Groups

We use Lemmas 1 and 2 to develop a fast search algorithm that finds the optimal partition of \mathcal{Y} for $p(x, y)$. The procedure is recursive, solving for optimal partitions on subalphabets in the solution of the optimal partition on \mathcal{Y} . For any alphabet $\mathcal{Y}' \subseteq \mathcal{Y}$, the procedure begins by making a list $\mathcal{L}_{\mathcal{Y}'}$ of all (single- or multi-level) groups that may appear in a partition \mathcal{P} of \mathcal{Y}' for $p(x, y)$ satisfying Lemmas 1 and 2. The list is initialized as $\mathcal{L}_{\mathcal{Y}'} = \{(y) : y \in \mathcal{Y}'\}$. Then for each symbol $y \in \mathcal{Y}'$ and each non-empty subset $\mathcal{S} \subseteq \{z \in \mathcal{Y}' : z \text{ can be combined with } y \text{ under } p(x, y)\}$ s.t. $\mathcal{L}_{\mathcal{Y}'}$ does not yet contain a group with elements $\mathcal{S} \cup \{y\}$, we find the optimal partition $\mathcal{P}_{\mathcal{S}}$ of \mathcal{S} for $p(x, y)$ and add to $\mathcal{L}_{\mathcal{Y}'}$ the group achieved by adding y to the empty root of $\mathcal{T}(\mathcal{P}_{\mathcal{S}})$ if $\mathcal{P}_{\mathcal{S}}$ contains more than one group or to the root of the single group in $\mathcal{P}_{\mathcal{S}}$ otherwise.

Partition Design

After constructing the above list of groups, we build a collection of partitions made of groups on that list. A few more tools are needed to describe the partition building algorithm. The following terminology is necessary for developing those tools.

We say that 1-level groups \mathcal{G}_1 and \mathcal{G}_2 (or nodes $\mathcal{T}(\mathcal{G}_1)$ and $\mathcal{T}(\mathcal{G}_2)$) can be combined under $p(x, y)$ if each pair $y_1 \in \mathcal{G}_1$, $y_2 \in \mathcal{G}_2$ can be combined under $p(x, y)$.

If $\mathcal{G}_I, \mathcal{G}_J \in \mathcal{P}$, so that \mathcal{G}_I and \mathcal{G}_J extend directly from the root \mathbf{r} of $\mathcal{T}(\mathcal{P})$ and nodes I and J are the roots of $\mathcal{T}(\mathcal{G}_I)$ and $\mathcal{T}(\mathcal{G}_J)$, and \mathcal{G}_o denotes the 1-level group at some node \mathbf{n}_o in $\mathcal{T}(\mathcal{G}_J)$, we say that \mathcal{G}_I can be combined with \mathcal{G}_J at \mathbf{n}_o if (1) I can be combined with \mathbf{n}_o and each of \mathbf{n}_o 's descendants in $\mathcal{T}(\mathcal{G}_J)$ and (2) \mathbf{n}_o and each of \mathbf{n}_o 's ancestors in $\mathcal{T}(\mathcal{G}_J)$ can be combined with I and each of I 's descendants in $\mathcal{T}(\mathcal{G}_I)$. The result of combining \mathcal{G}_I with \mathcal{G}_J at \mathcal{G}_o is a new group \mathcal{G}^* . Group \mathcal{G}^* modifies \mathcal{G}_J by replacing \mathcal{G}_o with 1-level group (I, \mathcal{G}_o) and adding the descendants of I (in addition to the descendants of \mathcal{G}_o) as descendants of (I, \mathcal{G}_o) in $\mathcal{T}(\mathcal{G}^*)$. Figure 1(d) shows an example where groups $\mathcal{G}_I = ((a_2) : \{(a_4), (a_5)\})$ and $\mathcal{G}_J = ((a_7) : \{(a_1), (a_3)\})$ of partition $\mathcal{P} = \{(a_0), \mathcal{G}_I, \mathcal{G}_J, (a_6)\}$ combine at (a_2) . The modified partition is $\mathcal{P}^* = \{(a_0), \mathcal{G}^*, (a_6)\}$, where $\mathcal{G}^* = ((a_2, a_7) : \{(a_1), (a_3), (a_4), (a_5)\})$.

The node probability $q(\mathbf{n})$ of a 1-level group $\mathbf{n} \in \mathcal{T}$ is the sum of the probabilities of that group's members. The subtree probability $Q(\mathbf{n})$ of the 1-level group at node $\mathbf{n} \in \mathcal{T}$ is the sum of probabilities of \mathbf{n} 's members and descendants. In Figure 1(b), let $p(\cdot)$ be Y 's marginal p.m.f., $q(23) = p(a_2)$ and $Q(23) = p(a_2) + p(a_4) + p(a_5)$.

Theorem 4 Let $\mathcal{P} = \{\mathcal{G}_1, \dots, \mathcal{G}_m\}$ be a partition of \mathcal{Y} under $p(x, y)$. Suppose that $\mathcal{G}_I \in \mathcal{P}$ can be combined with $\mathcal{G}_J \in \mathcal{P}$ at \mathcal{G}_o , where \mathcal{G}_o is the 1-level group at some node \mathbf{n}_o of $\mathcal{T}(\mathcal{G}_J)$. Let \mathcal{P}^* be the resulting partition. Then $El_{\mathcal{P}^*}^*(Y) \leq El_{\mathcal{P}}^*(Y)$.

Proof. Let $\mathbf{n}_o = Jj_1 \dots j_M = \mathbf{n}_p j_M$, so that \mathbf{n}_o 's parent is \mathbf{n}_p . Define $\mathcal{S}_1 = \{Jj_1 \dots j_i : 1 \leq i \leq M\}$; $\mathcal{S}_2 = \{\mathbf{n} \in \mathcal{T}(\mathcal{G}_J) : \mathbf{n} \text{ is the sibling of node } \mathbf{s}, \mathbf{s} \in \mathcal{S}_1\}$; $\mathcal{S}_3 = (\mathcal{S}_1 \cup \{J\}) \cap \{\mathbf{n}_o\}^c$. For any node $\mathbf{n} \in \mathcal{T}(\mathcal{P})$, let $Q_{\mathbf{n}}$ and $q_{\mathbf{n}}$ denote the subtree and node probabilities respectively of node \mathbf{n} in $\mathcal{T}(\mathcal{P})$, and define $\Delta Q_{\mathbf{n}} = Q_{\mathbf{n}} - q_{\mathbf{n}} = \sum_{j=1}^{K(\mathbf{n})} Q_{\mathbf{n}j}$.

Note that the sum of the subtree probabilities of \mathcal{G}_I and \mathcal{G}_J equals the subtree probability of \mathcal{G}^* , and thus the optimal average rate of the groups in $\mathcal{P} \cap \{\mathcal{G}_I, \mathcal{G}_J\}^c$ are not changed by the combination. Thus if (\bar{L}_I, \bar{L}_J) and $(\bar{L}_I^*, \bar{L}_J^*)$ are the optimal average rates for $(\mathcal{G}_I, \mathcal{G}_J)$ in \mathcal{P} and \mathcal{P}^* , respectively, then $\Delta \bar{L}_I + \Delta \bar{L}_J = (\bar{L}_I - \bar{L}_I^*) + (\bar{L}_J - \bar{L}_J^*)$ gives the total rate cost of changing from partition \mathcal{P} to partition \mathcal{P}^* . Here

$$\begin{aligned}
 -\bar{L}_I &= Q_I \log Q_I + \sum_{k=1}^{K(I)} Q_{Ik} \log \frac{Q_{Ik}}{\Delta Q_I} + \Delta l_I \\
 -\bar{L}_I^* &= Q_I \log \left((Q_I + Q_J) \prod_{\mathbf{n}k \in \mathcal{S}_1} \frac{Q_I + Q_{\mathbf{n}k}}{Q_I + \Delta Q_{\mathbf{n}}} \right) + \sum_{k=1}^{K(I)} Q_{Ik} \log \frac{Q_{Ik}}{\Delta Q_I + \Delta Q_{\mathbf{n}_o}} + \Delta l_I \\
 \Delta \bar{L}_I &= Q_I \log \left(\frac{Q_I + Q_J}{Q_I} \prod_{\mathbf{n}k \in \mathcal{S}_1} \frac{Q_I + Q_{\mathbf{n}k}}{Q_I + \Delta Q_{\mathbf{n}}} \right) + \sum_{k=1}^{K(I)} Q_{Ik} \log \frac{\Delta Q_I}{\Delta Q_I + \Delta Q_{\mathbf{n}_o}} \\
 &= Q_I \log \left(\frac{Q_I + Q_J}{Q_I + \Delta Q_J} \frac{Q_I + Q_{Jj_1}}{Q_I + \Delta Q_{Jj_1}} \dots \frac{Q_I + Q_{\mathbf{n}_p}}{Q_I + \Delta Q_{\mathbf{n}_p}} \frac{Q_I + Q_{\mathbf{n}_o}}{Q_I} \right)
 \end{aligned}$$

$$\begin{aligned}
& + \Delta Q_I \log \frac{\Delta Q_I}{\Delta Q_I + \Delta Q_{n_o}} \\
& = Q_I \log \prod_{n \in S_3} \frac{Q_I + Q_n}{Q_I + \Delta Q_n} + Q_I \log \left(1 + \frac{Q_{n_o}}{Q_I} \right) - \Delta Q_I \log \left(1 + \frac{\Delta Q_{n_o}}{\Delta Q_I} \right),
\end{aligned}$$

where Δl_I represents the portion of the average rate unchanged by the combination of \mathcal{G}_I and \mathcal{G}_J . It follows that $\Delta \bar{L}_I \geq 0$ since $\log \prod_{n \in S_3} (Q_I + Q_n) / (Q_I + \Delta Q_n) \geq 0$, and since $x \log(1 + c/x)$ is monotonically increasing in $x > 0$ and $c > 0$ implies that

$$\Delta Q_I \log \left(1 + \frac{\Delta Q_{n_o}}{\Delta Q_I} \right) \leq \Delta Q_I \log \left(1 + \frac{Q_{n_o}}{\Delta Q_I} \right) \leq Q_I \log \left(1 + \frac{Q_{n_o}}{Q_I} \right).$$

Similarly, using Δl_J as the portion of \bar{L}_J unchanged by the combination,

$$\begin{aligned}
-\bar{L}_J &= Q_J \log Q_J + \sum_{n \in S_1 \cup S_2} Q_n \log \frac{Q_n}{\Delta Q_n} + \sum_{k=1}^{K(n_o)} Q_{n_o k} \log \frac{Q_{n_o k}}{\Delta Q_{n_o}} + \Delta l_J \\
-\bar{L}_J^* &= Q_J \log(Q_J + Q_I) + \sum_{n \in S_1} Q_n \log \frac{Q_n + Q_I}{\Delta Q_n + Q_I} + \sum_{n \in S_2} Q_n \log \frac{Q_n}{\Delta Q_n + Q_I} \\
&\quad + \sum_{k=1}^{K(n_o)} Q_{n_o k} \log \frac{Q_{n_o k}}{\Delta Q_{n_o} + \Delta Q_I} + \Delta l_J \\
\Delta \bar{L}_J &= Q_J \log \left(\frac{Q_J + Q_I}{Q_J} \right) + \sum_{n \in S_1} Q_n \log \frac{Q_n + Q_I}{Q_n} + \sum_{n \in S_1} Q_n \log \frac{\Delta Q_n}{\Delta Q_n + Q_I} \\
&\quad + \sum_{n \in S_2} Q_n \log \frac{\Delta Q_n}{\Delta Q_n + Q_I} + \sum_{k=1}^{K(n_o)} Q_{n_o k} \log \frac{\Delta Q_{n_o}}{\Delta Q_{n_o} + \Delta Q_I} \\
&= Q_J \log \left(1 + \frac{Q_I}{Q_J} \right) + \sum_{n \in S_1} Q_n \log \left(1 + \frac{Q_I}{Q_n} \right) - \sum_{n \in S_3} \Delta Q_n \log \left(1 + \frac{Q_I}{\Delta Q_n} \right) \\
&\quad - \Delta Q_{n_o} \log \left(1 + \frac{\Delta Q_I}{\Delta Q_{n_o}} \right) \\
&\geq \sum_{n \in S_1 \cup S_3} \left[Q_n \log \left(1 + \frac{Q_I}{Q_n} \right) - \Delta Q_n \log \left(1 + \frac{Q_I}{\Delta Q_n} \right) \right]
\end{aligned}$$

Thus $\Delta \bar{L}_J \geq 0$ by the monotonicity of $x \log(1 + c/x)$. Since the optimal rates of \mathcal{G}_I and \mathcal{G}_J both decrease after combining, we have the desired result. \square

Theorem 5 Given partition \mathcal{P} of \mathcal{Y} on $p(x, y)$, if $\mathcal{G}_I, \mathcal{G}_J \in \mathcal{P}$ satisfy: (1) \mathcal{G}_I is a 1-level group and (2) \mathcal{G}_I can be combined with \mathcal{G}_J at root J of $\mathcal{T}(\mathcal{G}_J)$ to form partition \mathcal{P}^* , then $El_{\mathcal{P}^*}^{(H)}(Y) \leq El_{\mathcal{P}}^{(H)}(Y)$.

Proof. Let α denote the matched Huffman code for \mathcal{P} , and use α_I and α_J to denote this code's binary descriptions for nodes I and J . The binary description for any symbol in \mathcal{G}_I equals α_I ($\alpha(y) = \alpha_I$ for each $y \in \mathcal{G}_I$) while the binary description for any symbol in \mathcal{G}_J has prefix α_J ($\alpha(y) = \alpha_J \alpha'(y)$ for each $y \in \mathcal{G}_J$, where α' is a

matched Huffman code for \mathcal{G}_J). Let α_{\min} be the shorter of α_I and α_J . Since α is a matched Huffman code for \mathcal{P} and \mathcal{P}^* is a partition of \mathcal{Y} on $p(x, y)$,

$$\alpha^*(y) = \begin{cases} \alpha_{\min} & \text{if } y \in \mathcal{G}_I \\ \alpha_{\min} \alpha'(y) & \text{if } y \in \mathcal{G}_J \\ \alpha(y) & \text{otherwise} \end{cases}$$

is a matched code for \mathcal{P}^* . Further, $|\alpha_{\min}| \leq |\alpha_I|$ and $|\alpha_{\min}| \leq |\alpha_J|$ imply that the expected length of $\alpha^*(Y)$ is less than or equal to the expected length of $\alpha(Y)$ (but perhaps greater than the expected length of the matched Huffman code for \mathcal{P}^*). \square

Given the above results, we next recursively build the optimal partition of \mathcal{Y}' for $p(x, y)$. If any group $\mathcal{G} \in \mathcal{L}_{\mathcal{Y}'}$ contains all of the elements of \mathcal{Y}' , then $\mathcal{P}_{\mathcal{Y}'}^* = \{\mathcal{G}\}$ is the optimal partition on \mathcal{Y}' . Otherwise, the algorithm systematically builds a partition, adding one group at a time from $\mathcal{L}_{\mathcal{Y}'}$ to set \mathcal{P} until \mathcal{P} is a complete partition. For $\mathcal{G} \in \mathcal{L}_{\mathcal{Y}'}$ to be added to \mathcal{P} , it must satisfy: (1) $\mathcal{G} \cap \mathcal{G}' = \emptyset$ and (2) $\mathcal{G}, \mathcal{G}'$ cannot be combined (see Theorem 4 for arithmetic or Theorem 5 for Huffman coding) for all $\mathcal{G}' \in \mathcal{P}$. For each complete partition, we find the rate of the optimal code on \mathcal{P} . The optimal partition is the partition whose optimal code gives the lowest expected rate. A lower complexity higher memory algorithm is achieved by recursively building optimal matched codes for the partial partitions and ruling out partial partitions for which another partial partition on the same alphabet yields a lower rate.

III Lossless and Near Lossless Instantaneous MASCS

The generalization of the lossless side-information code first to lossless general MASC coding and then to near lossless side-information and MASC coding follows.

Instantaneous Lossless Multiple Access Source Coding

We here drop the side-information coding assumption that X (or Y) can be decoded independently. The solution of the resulting generalized instantaneous lossless MASC problem requires two partitions, one on \mathcal{X} and one on \mathcal{Y} , for $p(x, y)$. As in side-information coding, these partitions, denoted by \mathcal{P}_X and \mathcal{P}_Y , describe the prefix and equivalence relationships in the binary descriptions of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Thus every instantaneous lossless MASC can be described as a matched code on \mathcal{P}_X and a matched code on \mathcal{P}_Y for some $(\mathcal{P}_X, \mathcal{P}_Y)$. Since optimality of a matched code for partition \mathcal{P} is independent of whether \mathcal{P} is used in a side-information code or an MASC, our optimal matched code design methods from Section II apply here as well.

The problem that remains is partition design. To be employed in a lossless MASC, both \mathcal{P}_X and \mathcal{P}_Y must satisfy all of the conditions of a partition used in lossless side-information coding. (If \mathcal{P}_Y fails to uniquely describe Y when the decoder knows X exactly, then it also fails when the decoder knows X imperfectly. The corresponding statement holds for \mathcal{P}_X .) The argument that all non-root nodes can be non-empty in the optimal partition again holds, and thus we restrict our attention to such partitions.

For $(\mathcal{P}_X, \mathcal{P}_Y)$ to yield an *instantaneous* MASC, the decoder must recognize when it has reached the end of $\gamma_X(X)$ and $\gamma_Y(Y)$. The decoder proceeds as follows. We think of a matched code on \mathcal{P} as a multi-stage description, with each stage corresponding

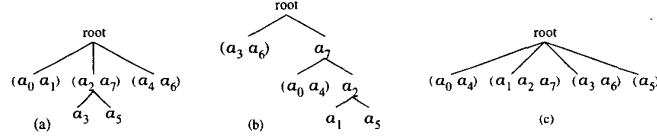


Figure 2: Partition trees for (a) $R_{SI,A}^*(Y)$; (b) $R_{SI,H}^*(Y)$; (c) $R'_{SI,A}(Y)$ and $R'_{SI,H}(Y)$.

to a single level in $\mathcal{T}(\mathcal{P})$. Starting at the roots of $\mathcal{T}(\mathcal{P}_X)$ and $\mathcal{T}(\mathcal{P}_Y)$, the decoder reads the first-stages of $\gamma_X(X)$ and $\gamma_Y(Y)$, traversing the path from the root to a (non-empty) node in each partition. If these nodes $(\mathbf{n}_X, \mathbf{n}_Y)$ have no descendants, then the decoder has reached the end of $\gamma_X(X)$ and $\gamma_Y(Y)$. Otherwise, let $(\mathcal{T}_X, \mathcal{T}_Y)$ be the subtrees with roots $(\mathbf{n}_X, \mathbf{n}_Y)$ respectively. For instantaneous coding, one of the following must hold (A) $X \in \mathcal{T}_X$ implies that $Y \in \mathbf{n}_Y$ and $Y \in \mathcal{T}_Y$ implies that $X \in \mathbf{n}_Y$; (B) $X \in \mathcal{T}_X$ implies that $Y \notin \mathbf{n}_Y$; or (C) $Y \in \mathcal{T}_Y$ implies that $X \notin \mathbf{n}_X$. Under condition (A), the decoder stops decoding $\gamma_X(X)$ and $\gamma_Y(Y)$. Under condition (B), the decoder reads the next stage of $\gamma_Y(Y)$ and traverses the described path in $\mathcal{T}(\mathcal{P}_Y)$ to node \mathbf{n}'_Y . Condition (C) similarly leads to a new node \mathbf{n}'_X . The procedure repeats on succeeding node pairs until decoding of both $\gamma_X(X)$ and $\gamma_Y(Y)$ are completed.

Let $(\mathcal{G}^{(X)}, \mathcal{G}^{(Y)})$ be the 1-level groups described by $(\gamma_X(X), \gamma_Y(Y))$. Then $(\mathcal{P}_X, \mathcal{P}_Y)$ gives a *lossless* MASC if there is exactly one $(x, y) \in \mathcal{G}^{(X)} \times \mathcal{G}^{(Y)}$ with $p(x, y) > 0$.

For every partition \mathcal{P}_X , a variation on the partition search algorithm of Section II finds the best \mathcal{P}_Y for which $(\mathcal{P}_X, \mathcal{P}_Y)$ yields an instantaneous lossless MASC [7]. Traversing all \mathcal{P}_X allows us find all partitions with performances on the convex hull of the achievable rate region. Reversing the roles of X and Y yields identical results.

Near Lossless Instantaneous Multiple Access Source Coding

For any small error probability P_e , we wish to find an MASC that decodes correctly at least proportion $1 - P_e$ of the time. For each $\mathcal{S}_e \in \{\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y} : \sum_{(x,y) \in \mathcal{S}} p(x, y) \leq P_e\}$, a lossless MASC on the p.m.f. obtained by zeroing out $p(x, y)$ on \mathcal{S}_e and renormalizing is a lossless MASC on $p(x, y)$. The achievable rate region is the union of the rate regions associated with all such \mathcal{S}_e .

IV Experimental Results

This section shows experimental results for the algorithms introduced in Sections II and III. All experiments use the p.m.f. described in Section II.

Let $H(X)$ and $R_H(X)$ be the optimal and Huffman traditional source coding rates on X ; and use $[H(Y), R'_{SI,A}(Y), R^*_{SI,A}(Y)]$ and $[R_H(Y), R'_{SI,H}(Y), R^*_{SI,H}(Y)]$ to denote the optimal and Huffman results respectively for [traditional, [3] side-information, and our side-information] coding on Y . A summary of our results follows.

| $H(X)$ | $R_H(X)$ | $H(Y)$ | $R'_{SI,A}(Y)$ | $R^*_{SI,A}(Y)$ | $R_H(Y)$ | $R'_{SI,H}(Y)$ | $R^*_{SI,H}(Y)$ |
|---------|----------|---------|----------------|-----------------|----------|----------------|-----------------|
| 2.91412 | 2.97 | 2.91075 | 1.67976 | 1.53582 | 2.96 | 1.75 | 1.67 |

Figure 3 compares traditional and MASC rate region results. Setting $P_e = 0.04$ gets us to the Slepian-Wolf bound for this example.

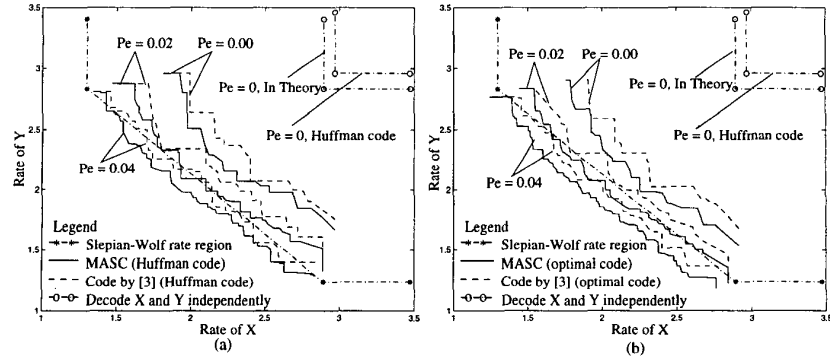


Figure 3: Rate regions (a) Huffman and (b) optimal codes.

V Summary

In this paper, we treat the practical lossless and near lossless source coding problems for general multiple access networks with arbitrary source p.m.f. $p(x, y)$, giving a constructive and efficient code design algorithm. Our experimental results based on this algorithm are consistent with the theory of MASC and demonstrate its feasibility in optimal code design.

References

- [1] D.Slepian and J.K.Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, IT-19(4):471–480, July 1973.
- [2] H. S. Witsenhausen. The zero-error side information problem and chromatic numbers. *IEEE Transactions on Information Theory*, 22:592–593, 1976.
- [3] A. Kh. Al Jabri and S. Al-Issa. Zero-error codes for correlated information sources. In *Proceedings of Cryptography*, pages 17–22, Cirencester, UK, December 1997.
- [4] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS) design and construction. In *Proceedings of the Data Compression Conference*, pages 158–167, Snowbird, UT, March 1999. IEEE.
- [5] Y. Yan and T. Berger. On instantaneous codes for zero-error coding of two correlated sources. In *Proceedings of the IEEE International Symposium on Information Theory*, page 344, Sorrento, Italy, June 2000. IEEE.
- [6] Q. Zhao and M. Effros. Lossless source coding for multiple access networks. Submitted to ISIT 2001.
- [7] Q. Zhao and M. Effros. Lossless and near lossless source coding for multiple access networks. In preparation for submission to IEEE Trans. IT.